DOCUMENT RESUME

ED 285 567 IR 012 773

AUTHOR Winn, William
TITLE Real-Time Graphics for CAI: A Rudimentary Grammar and Demonstration Program.
PUB DATE Feb 87
NOTE 23p.; Paper presented at the Annual Convention of the Association for Educational Communications and Technology (Atlanta, GA, February 26-March 1, 1987). For the complete proceedings, see IR 012 723.
PUB TYPE Information Analyses (070) -- Reports - Research/Technical (143) -- Speeches/Conference Papers (150)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS *Computer Assisted Instruction; *Computer Graphics; *Expert Systems; Instructional Design; Man Machine Systems; *Models
IDENTIFIERS AECT Research and Theory Division Meeting

ABSTRACT

This paper focuses on graphics and how they can be created, in real time, from information stored in a database, and the application of this technique to computer-assisted instruction (CAI). It is noted that this is a special case of the general trend towards endowing instructional systems with a degree of decision-making or design expertise, as it requires that the system not only present information in a variety of graphic formats to satisfy the requirements of a large repertoire of instructional strategies, but also that the system decide upon the format in which information is to be presented during instruction. A generalized model of an intelligent CAI system composed of a database, a collection of rules, and an interface is discussed, as well as a grammar for graphics which represents the decision rules of a linguistic model of such a system. The three components of the grammar are also discussed, i.e., syntax, semantics, and pragmatics. A program which allows the user to interact with a database and retrieve information in either textual or graphic formats is then described to illustrate how a grammar of graphics might operate. Diagrams of the intelligent CAI, linguistic, and program models are appended, and 50 references are listed. (MES)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED285567

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

X This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

Title:

Real-Time Graphics for CAI: A Rudimentary Grammar and Demonstration Program

Author:

Bill Winn

IR012773

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
Michael Simonson

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

BEST COPY AVAILABLE

Real-Time Graphics for CAI: A Rudimentary Grammar and Demonstration Program.

Bill Winn

University of Washington.

INTRODUCTION

The ideas presented in this paper began with the realization that in intelligent computer-assisted instruction (ICAI) systems it is possible to make decisions about instructional strategies while instruction is underway. Instructional design has evolved in a context where the selection of instructional strategies is separated in time from their implementation (Richie, 1986; Winn, 1986a). This has caused a number of difficulties (Nunan, 1983) concerned primarily with the ability of the designer to predict what will occur once instruction has started. If instructional design is concerned with making decisions based in part upon the conditions that prevail when instruction takes place (Reigeluth, 1983), and if as is usually the case those conditions change as a result of instruction occurring, then the difficulty of the designer's task becomes apparent.

There are two alternatives towards which the designer might strive in order to avoid this difficulty (Winn, in press). The first is the creation of a complete and predictively valid theory of instruction that is capable of predicting with considerable accuracy all of the contingencies that might arise during instruction. Instructional theory is at present a long way away from this ideal. The second alternative is to endow the instructional systems through which instruction is implemented with sufficient "expertise" to deal with problems on the spot in much the same way as an expert teacher can (see Berliner, 1986; Schon, 1983). This is presently the more realistic alternative because ways in which artificial intelligence can be brought to bear upon instructional problems are being developed (Sleeman and Brown, 1982).

An important aspect of building expertise in instructional decision-making into ICAI systems has to do with the ways in which information is presented on the screen. Granted, information presentation is only one part of instruction. (Gagne and Briggs [1979] list nine "events" that should occur during instruction, of which presenting information is only one). However, the need for the system to contain a large pool of potentially useful methods for selection through the application of prescriptive instructional principles requires that the system also be capable of presenting information in a variety of textual and graphic formats. This is because different learning tasks for learners of different abilities

and styles optimally require information to be presented in formats that match those tasks and abilities. The body of literature into individual differences in ability to learn different tasks from different presentation formats is well known. Also well known is the fact that one-on-one tutoring, of the kind that ICAI can provide, leads to remarkable gains in achievement when compared to more traditional modes of instruciton (Bloom, 1984). Successful tutoring requires that information be presented in consideration of students' individual differences in information processing.

Students typically interact with computers through interfaces built around language. This is probably because the development of artificial intelligence has been closely coupled to research into a number of aspects of language, such as semantics, syntax, how information is stored in memory, and how comprehensible utterances are produced (Norman and Rumelhart, 1975; Schank, 1984; Schank and Abelson, 1977). As a result, the capability of intelligent systems to present information in graphics has been less well developed (Christopher, 1985), with some notable exceptions, such as STEAMER (Hollan, Hutchins and Weitzman, 1984), which is an expert system, using intelligent graphics, that teaches about steam power plants in ships.

The focus of this paper is on graphics and how they can be created, in real time, from information stored in a database. This is a special case of the general trend towards endowing instructional systems with a degree of decision-making, or design, expertise. It requires that the system not only present information in a variety of graphic formats to satisfy the requirements of a large repertoire of instructional strategies, but also that the system decide upon the format in which information is to be presented during instruction. This would provide the equivalent of a teacher drawing a chart or diagram on the blackboard on the spur of the moment in order to clarify a particular point or elaborate upon a particular concept.

The term "graphics" is used in this paper to refer to illustrations whose purpose is to make abstract ideas more concrete through the use of spatial analogy. These illustrations stand in contrast to pictorial illustrations whose purpose is often to make very concrete ideas more abstract (see Dwyer, 1978; Winn, 1986b). Operationally, this means that the paper is talking primarily about diagrams and charts, although allusion will be made to graphs as well as the need arises.

A MODEL FOR ICAI

Figure 1 illustrates a generalized model of the way an ICAI system might be constructed. It has the following characteristics:

==========
Figure 1 about here
==========

1. Information it "knows about" is stored in a database in an abstract form, such as a semantic network. The criterion is that the propositions in the database must be capable of expression in any of a number of textual or graphic formats, hence the need for them to be expressed in abstract form.
2. The student interacts with the database through a two-way interface consisting of a number of parts. For input, the interface consists of a parser that determines whether the student's input is something that the system can recognize, and a set of diagnostic routines that determine what it is the student needs to know and how the student can best be serviced. The output side of the interface takes information derived from decisions made by a set of decision rules and converts that information into screen displays.
3. The heart of the system is a collection of rules that operate in accordance with a number of factors. The first of these consists of the instructional conditions that currently prevail, such as what the student has already learned, the student's mental model of the material to be mastered and how correct it is (see Anderson, Boyle and Yost, 1983; Larkin, 1985), the student's abilities and learning styles. Second is the information that has to be presented, that has been retrieved from the database. And third is the syntax of the forms in which information is to be presented. Thus, the decision rules can be exemplified in the following:

> Given that the student has already learned what a mammal is and is a poor verbal learner; (conditions)
> And that the information to be presented is contained in the proposition "whale ISA mammal"; (information)
> And that, in diagrams, superordinate concepts are placed above subordinate concepts; (syntax)
> Then place "mammal" above "whale" and join it to "whale" with a vertical line labelled to indicate class membership (strategy).

Rules such as these clearly draw upon theory in a number of different areas. Obviously, instructional theory containing prescriptive principles involving instructional conditions, outcomes and methods (Reigeluth, 1983), is involved. In addition, there are syntactic rules that specify the relative placement on the screen of subordinate and superordinate concepts. And there are semantic rules that map the information from the database onto the structures specified by the syntax. In other words, there is sufficient information in the proposition "whale ISA mammal" for the system to determine which concept is subordinate to which. In order to examine the way in which these two latter types of rules operate, it is useful to shift our context to a linguistic one for the moment, and look at what a "grammar" of graphics might involve.

GRAMMAR

Figure 2 represents the same general model for ICAI but as a

linguist might conceive of it. A number of features of the model remain unchanged. But the decision rules are now thought of as a "grammar" which, as grammars do, has three components to it: a syntactic component, a semantic component and a pragmatic component. (See Salomon, 1979, for a discussion of syntactic and semantic aspects of symbol systems and Kosslyn et. al., 1983, for an elaboration of how these three components function in relation to the meaning of graphs). Each of these will now be examined.

Syntax.

A syntax can be thought of as a set of rules that permit the creation of all the acceptable structures, and only acceptable structures, that the utterances of a language can have. Syntactic structures are content-independent, and thus syntax is considered to be separate from semantics (Chomsky, 1962, 1965; Fowler, 1971). Theories of syntax are derived from the study of syntax of language. While it may not be legitimate to draw parallels between language and graphics for all aspects of grammar, we take the construct of syntax, as developed for language, as the basic paradigm, and diverge from it where necessary.

Structure. The syntactic structure of graphics consists of two types of fundamental component which may behave in certain ways and which are governed by sets of rules analogous to phrase structure rules. The two components are "elements" and "relationships", (see Knowlton, 1966; Szlichsinski, 1980; Schlichtmann, 1980 among others for a more detailed exploration of this idea).

Elements consist of concepts or clusters of concepts. "Concept" is used here in the same sense as it is used by Norman and Rumelhart (1975, pp. 40-44) when describing active semantic networks. Concepts are what propositions make statements about and between which predicates describe relationships. In a tree diagram of animals, the labels "cat", "elephant", "mammal", "vertebrate", referring to these particular concepts, would correspond to "elements" in the syntax.

Relationships describe how concepts are connected. They correspond to predicates in semantic networks. In any syntactic system, they are finite and usually small in number, while the number of concepts and thus the number of potential utterances is in theory unlimited. In the tree diagram of animals, relationships between pairs of concepts would represent predicates indicating class membership (cat ISA mammal), the attribution of qualities (elephant IS big), of characteristics (elephant HAS trunk), and so on.

Elements and relationships can be thought of as variables, or placeholders, that are instantiated when the semantic component is added to a structure. In this sense, structures can be thought of as abstract schemata that are used for the interpretation of information (Anderson 1977; Neisser, 1976; Thorndyke & Hayes-Roth, 1979). However, when we keep the syntactic component separate from the semantic component, all we can do is identify the dimensions

along which the elements and relationships can vary.

Elements can vary along two dimensions, notationality and dominance.

1. Notationality (Goodman, 1968; Salomon, 1979) refers to the symbol systems that are used when an element is represented in a graphic. In notational systems, each symbol is discrete and separable from others. In a non-notational system, this is not the case. Where one symbol begins and another ends is not easily determined. Thus, in the animal hierarchy, the concepts could be represented notationally as labels or non-notationally as pictures. A critical correlate of notationality is "repleteness". This alludes to how many dimensions in the representation of a message are meaningful. In a color photograph, size, color, shape, form, placement, contrast, etc., all carry meaning. In a flow diagram only position of the elements on the page and the arrows might be meaningful. The thickness of the lines, whether they are colored or not and even whether the elements are shown as pictures or labels might not be relevant. Salomon (1979) goes into this in much more detail.

2. Dominance refers to those characteristics of the way in which an element is represented that make it more or less dominant to the human perceptual system. Thus, the size with which elements are represented, and the way they are made to stand in contrast to other elements by color cueing, different styles of type, and so on, affect dominance.

Relationships can vary along three dimensions, proximity, placement, and type of link. Placement itself involves a number of additional dimensions.

1. Proximity refers to how close one element is to another on the page or screen. As the proximity among groups of elements varies, they form more or less distinct clusters of elements. It should be noted that well-defined clusters of elements can function in the same way as single elements in a structure, thus enabling the formation of hierarchies and subsets of elements.

2. Placement can refer to the placement of elements relative to the frame (screen or page boundaries) or to each other. These two dimensions are referred to as absolute and relative placement respectively.

i. Absolute placement is concerned with whether an element is at the top, bottom, left or right or middle of the frame.

ii. In the case of relative placement, an element can be above, below, to the left or right of another, or can be completely or partly enclosed within another. The former is illustrated when the word "mammal" is placed above "cat" to indicate class membership, the latter when "cat" is included with "dog" and other labels within a circle labelled "mammal".

3. Links. The links among elements may be inferred from proximity and placement, or they may be made explicit by means of a number of devices. These include representations of predicates, such as the label "belongs to" or "has", lines, arrows, and combinations of these. The notationality and the repleteness of the links can

vary in ways that were discussed above for elements for elements.

Syntactic Rules.

Syntactic structures for graphics are derived from rules that specify how the components of a structure may vary along the dimensions we have just described. (If a structure is considered to be analogous to a sentence, then these rules operate in the same way as phrase structure rules in transformational grammar. For example, NP -> Art + N is a legal rule, while NP -> N + Art is not, at least for English). Some of the syntactic rules of graphics are:

1. Elements that are causes of or antecedent to other elements are placed to their left and/or joined to them by arrows pointing to the effect or object.
2. Elements representing superordinate or inclusive classes are placed above elements representing subordinate or included concepts.
3. Subordinate or included concepts are placed inside elements representing superordinate classes (e.g. set and Venn diagrams).
4. Qualities and attributes of a concept are "attached to" the element that represents it. If the element is represented using a non-notational symbol system, the qualities and attributes may be represented as pictures. If a notational system is used, the attachment of qualities and attributes may be achieved by placing labels describing them close to the element, or by including them in the same box or circle that encloses the label for the element.
5. Elements that are more important than others should be made dominant.
6. Elements representing concepts that are conceptually related should be placed close together.

Psychological Validity.

It should be noted that a number of these rules appear to be statements of the obvious. The reason for this is that any syntax is first and foremost descriptive of usage rather than prescriptive. Some of the rules are therefore derived from the conventions of a particular usage of graphics for communication that is often culturally based. For example, rule one requiring that causes be placed to the left of their effects describes common usage in those cultures where language is written from left to right across the page. Other rules have validity on the basis of psychological research and theory.

Kosslyn et al. (1983) remind us that the Gestalt psychologists (Wertheimer, 1938) developed a number of rules pertaining to the ways in which elements are placed and linked in a display may be meaningfully linked. (Fleming and Levie, 1978, also describe a number of these rules as "principles" of message design). Wertheimer's (1938) laws include: Proximity, which states that elements placed close together tend to be associated; Similarity, stating that elements that look alike tend to be associated; Symmetry, which tends to impose figural cohesion onto collections of elements; Good continuation, which claims that when elements form

good figures in their aggregate they will be seen as belonging together; Common fate, which ascribes a similar quality to apparently moving elements that have a common trajectory; Good figure, in which elements having a closed boundary are more likely to be seen as wholes than figures lacking this feature.

In addition to Gestalt principles, other psychological studies of graphics and how people process the information they contain have have confirmed the validity of our rules of graphic syntax. Winn (1982, 1983) has shown that elements in flow diagrams tend to be processed from left to right and top to bottom, suggesting that antecedents should be placed to the left of causes and superordinate categories above subordinate ones. Rabinowitz and Mandler (1983) found that simply placing collections of words to be learned serially in physical groups on a page improved subjects' ability to remember them. In a summary of a number of studies, Hartley (in press) has reported that the way in which text is placed on the page affects the way it is interpreted. And a number of researchers (Holliday, Brunner and Donais, 1977; Koran and Koran, 1980; Winn and Holliday, 1982) have reported that flow diagrams complying with various of the syntactic rules described above have made it easier for students to learn material.

Semantics.

The semantic component of the grammar is concerned with the way in which information is made meaningful by mapping it onto a structure. It is assumed that the information is contained in propositions in a database of some sort, and that these propositions contain information as concepts and predicates that can be mapped onto a structure. The semantic rules of the system must achieve several purposes.

1. They must link specific predicates with specific syntactic rules. That is to say, they must lead to the identification of the predicate ISA with subordination, and with the syntactic rule that requires subordinates to be placed below superordinates. Or they must associate CAUSE with cause-effect relationships and the rule that places causes to the the left of effects.

2. They must be able to associate different concepts with the appropriate placeholders in the structure. For instance, "mammal" must be identified as the concept that is superordinate to "cat" in the proposition "cat ISA mammal". In the same way, "lightning" must be identified as the cause in the proposition "lightning CAUSE fire".

3. They must be able to determine which concepts from the database are conceptually "close", and thus to identify clusters of concepts. Therefore, they must be able to determine how many concepts or "nodes" in the network occur between any two concepts. If "cat" and "elephant" have only one node between them, "mammal", and "snake" and "lizard" have only one, "reptile", between them, with "mammal" and "reptile" being linked by an additional node, "animal", then "cat" and "elephant" are closer to each other than to the two concepts in the "reptile" category and form a cluster.

4. They must be able to determine which concepts are dominant over others, to identify those that act as "hubs" around which others can be said to "revolve". Thus, if the network contains a number of propositions each linking the concept "cat" through a variety of predicates to a number of other concepts, then "cat" can be said to be dominant.

The information that enables the derivation and correct application of these rules is within the propositional network itself. This presupposes that the information in the database is in fact correct. For example, it is conceivable that a network contain the proposition "mammal ISA cat". If the information in the network is not correct, then the semantic component of the grammar must contain additional rules that check the propositions against other criteria in order to determine their veracity.

Pragmatics.

Generally, pragmatics is concerned with the behavioral effects of human communication (Watzlawick, Beavin and Jackson, 1967). We shall consider pragmatics in the narrower sense, following Kosslyn et. al. (1983), of the inferences that are drawn from a message and the context in which it occurs. It is here that the discrepancies occur between the intended meaning of semantically correct messages and what they actually mean to the receiver. The rules concerned with the pragmatic component of the grammar therefore deal with what inferences are likely to be made and the context in which the rules are being brought to bear. If we return for a moment to the idea of instructional design, presented right at the beginning of the paper, then we can see a parallel between the pragmatic component and the "conditions of instruction". The inferences that a person is likely to draw from a message are determined by a great variety of conditions (Salomon, 1981). These have to do with factors such as how much the person already knows about the conceptual domain in question (Tobias, 1976), the persons's aptitudes and mental skills (Cronbach and Snow, 1977), belief systems (Schoenfeld, 1983), perception of how much effort will have to be invested in processing the information (Salomon, 1982, 1983), and perception of the likelihood of success (Schunk, 1984). Collectively, these factors can be thought of as the context of the communciation.

A great deal has been written about how all of these factors affect how people interpret messages, and about how messages should be constructed so as to minimize discrepancies between intended and perceived meaning that can be traced to these factors. These will not be presented here. Suffice it to say that we are here concerned with what has been addressed in a number of descriptive and prescriptive theories of instruction, which have been described in Reigeluth (1983), Gagne and Dick (1983), and elsewhere.

IMPLEMENTATION

In order to illustrate how a grammar of graphics, comprising syntactic, semantic and pragmatic rules, might operate, a program has been written that allows a user to interact with a database and to retrieve information from it in either textual or graphic formats. This program will now be described. In so doing, those aspects of the general model of ICAI not dealt with in much detail so far, the input parser and display routines, will be described in more detail. It is important to remember that the program was written to demonstrate how the various components might interact, and also that it does not have an instructional component. It simply allows a user, who might be a student, to interrogate a database and to retrieve information from it. In other words, it is concerned primarily with syntactic and semantic rules, and simply responds to requests from the user without diagnosing the user's needs or adapting output to suit the user's prior knowledge, styles, aptitudes, or perceptions. However, it is expected that an instructional component could be added to a system such as this, and that the principles demonstrated in the program would hold true for instructional applications.

Figure 3 illustrates the same general model as figures 1 and 2 and shows the particular details of the model as implemented in the program.

==========
Figure 3 about here
==========

Database.

The database consists of a semantic network (Norman and Rumelhart, 1975) made up of concepts (nodes) and predicates that describe how they are related. At present, the database may contain seventy concepts. A concept may be a member of a class of concepts (an elephant is a mammal), and may have other concepts subordinate to it (Fred is an elephant). A concept may have attributes (for example, an elephant has a trunk), and it may have qualities (an elephant is large). The location of a concept can be specified (the elephant is in the zoo). A concept can act (the elephant trumpets). It can act on another concept (the elephant eats hay), or it can bring about a particular effect (an elephant causes fear). In the database, concepts (elephant, mammal, Fred), as well as attributes, qualities, locations, actions, objects of actions and effects (trunk, large, zoo, trumpet, hay and fear) are all treated as concepts. This allows the easy construction of large hierarchies of classes and extended cause-and-effect chains. It also allows the construction of equivalents of "Object-Attribute-Value triplets" (Harmon and King, 1985), with constructions such as, "The cat has fur; the fur is white".

The database is not content-specific. It can contain any concepts the user wishes to place in it, and is created interactively by the user.

There are six predicates that describe how the concepts in the database are related:

1. ISA describes class membership (elephant ISA mammal).
2. HAS attaches attributes to a concept (elephant HAS trunk).
3. IS ascribes qualities to concepts (elephant IS large).
4. DOES ascribes actions to concepts (elephant DOES trumpet, elephant DOES eat hay).
5. LOC describes where a concept is located (elephant LOC zoo).
6. CAUSE establishes a cause-and-effect relationship between concepts (elephant CAUSE fear).

The database is organized as a matrix, with the concepts as row and column headings and the predicates, numbered one through six in the order of the list above, in the appropriate cells. If "elephant" were concept three and "mammal" concept nine, then the proposition "elephant ISA mammal" would be represented by placing a "1" in the cell where the third line intersected the ninth column. Negatives are stored as negative numbers, and when the DOES predicate involves an object of the specified action, the predicate is represented as 100 added to the number of the concept that is the object of the action. Thus with "elephant" as concept three, "eat" as concept 16, "hay" as concept nine, the proposition "elephant DOES not eat hay" would be represented by placing the number -109 at the intersection of the third and sixteenth columns, (minus for "not", 100 for DOES with object, plus 9 for "hay"). (Representing the information in the database in this way is appropriate for a program written in BASIC. With other languages, e.g. LISP or PROLOG, other structures would be more appropriate). By scanning along a row in the matrix, it is therefore very simple to retrieve a full description of any concept in the database, or to answer specific questions about it.

Program.

The program performs three main functions: database creation, database interrogation and database management. The last of these, involving the editing of the database, saving and loading database files, help, and so on, is straightforward and will not be discussed. The first will be described briefly, and the second at more length.

Database creation.

The interface contains a simple parser that interprets quasi-natural English statements. The user adds to the database by typing statements such as "A cat is an animal", "The elephant trumpets", "The elephant does not eat hamburger". The parser checks to see if the syntax of the statement is correct. If so, it looks to see if

the concepts are already in the database. If not, it adds them. It then places the number representing the predicate in the appropriate location in the matrix.

Database Interrogation.

The parser also processes users' questions. Questions fall into two categories: those requiring yes or no answers, and those requiring constructed answers.

For the first category of questions, the parser treats predicate names as interrogatives. If the user asks "Does a cat drink milk?", the parser checks the syntax, and then sees whether "cat", "drink" and "milk" are in the database. If not, it tells the user. It then checks along the "cat" row of the matrix to see whether the predicate "DOES" (100) added the concept number for "milk" is in the "drink" column. It replies "Yes" or "No" accordingly".

For constructed questions, the parser recognizes "What" and "Which" as interrogatives. It responds to questions such as "What is a cat?", "What does an elephant eat?" by looking up the answer in the matrix, using the appropriate line for the concept named in the question. When the interrogative is "Which", the procedure is exactly he same, except it scans the column, not the row, corresponding to the named concept. Thus "What is an elephant?" yields "mammal", and "Which is an elephant?" gives "Fred". Similarly, "What does an elephant eat?" gives "hay", and "Which does eat the elephant?" would give, say, "lion".

The parser recognizes other interrogatives. "Describe" gives a complete description of the named concept. "Vocab" gives the list of concepts in the database. "Map" displays a map of the database. And "expert" engages a simple expert system which asks the user questions about a concept in the database in an attempt to guess what it is.

This brief description has assumed that the program has displayed text on the screen in response to questions. The program also operates in graphics mode, allowing the results of certain interrogations to be displayed as diagrams.

Graphics Mode.

All of the commands available in text mode are available in graphics mode. Graphics mode differs by providing some additional commands for zooming in and out on an image and for moving an image around the screen. It also differs in that it provides graphic answers to "what" and "which" interrogations. Thus, if the user types in "what is an elephant", the word "elephant" appears in a box beneath a box containing the word "mammal" to which it is joined by a vertical line labelled "is a". In response to "what does an elephant eat", a third box is added to the right of the elephant box containing the word "hay", joined to "elephant" with a line labelled "eat". At present, the graphics consist of labels, boxes and straight lines. However, there is no reason why concepts might not be represented by line drawings or icons of some sort.

The program contains a small grammar that allows it to display these graphics. The grammar consists of simple rules that embody some of the principles that were discussed in an earlier section. Bearing in mind that what the user sees on the screen is labels for predicates, concepts and their attributes, not pictures, these rules are:

1. Subordinate concepts are placed in boxes beneath the superordinate category to which they belong. (If more than one concept is subordinate to another, then it may be offset to the left or right of the position vertically below the superordinate category.)
2. Causes are placed to the left of their effects, agents to the left of their objects.
3. Predicates are expressed as labelled lines joining the two concepts whose relationship they describe. The labels for predicates are not placed in boxes.
4. The attributes and qualities of concepts and other characteristics, such as their location, are placed within the boxes that contain the concepts' labels.

In graphics mode, the parsing of input is identical to the way it is done in text mode. Thus the user can add to, edit and interrogate the database as before. Also, for all interrogations, other than "what" or "which", the answer is printed as text at the top of the screen. However, if the parser encounters "what" or "which" in a syntactically correct interrogation, the following processes occur:

1. If the concept named in the question is already on the screen, the program moves to step 2. If not, the screen is cleared and the concept is placed in its box in the middle of the screen.
2. The program seeks for the appropriate response. However, because it must now display a single concept, not a list, it must find the "closest" concept to the one named in the question. For example, an elephant is a mammal and an animal. In text mode, both would be provided as answers to "What is an elephant?". In graphics mode, only "mammal" must be displayed. Thus the program finds all possible responses in the appropriate row of the matrix, and then checks to see which is subordinate to which. The concept that has nothing subordinate to it except the concept named in the question is selected as the correct answer. The exceptions to this are answers to "which is a ..." questions, where any number of subordinate concepts can be displayed provided that they are all immediately subordinate to the first and all on the same "level" in the hierarchy.
3. The four rules described above are then applied. There are five possible outcomes:
   i. If the interrogative is "what" and the predicate is ISA, then the response will be placed above the concept named in the question.
   ii. If the interrogative is "what" and the predicate is DOES, the reponse will be placed to the right.
   iii. If the interrogative is "which" and the predicate is ISA,

then the response(s) will be placed below the named concept.

iv. If the interrogative is "which" and the predicate is DOES, then the response will be placed to the left.

v. If the interrogative is "what" and the predicate is one of IS, HAS, or LOC, then the response will be placed in the box beneath the concept's name.

4. The scale at which the diagram is to be drawn is then checked. If the user has "zoomed out" in order to be able to see more information on the screen at once, certain restrictions to the display apply. As the user "zooms out", the size of the boxes, and therefore the amount of information that they can contain, is reduced. After the first zoom out, only concept names can be included in the boxes, not attributes, values, locations and so on. Also, predicate labels cannot be displayed beside the lines joining the concepts. If the user zooms out further, then even the concept labels are lost. (Zooming in again restores the labels). Thus zooming out reduces the amount of detail about each concept that is presented, but improves the view of the whole pattern of the diagram.

5. The concept label in its box is added to the diagram on the screen. If the interrogation is "which is a ...", and there is more than one answer, then the concept labels in their boxes are drawn equally spaced side by side below the concept in the question.

6. Using information it has obtained in step three, the program determines which sides (top, bottom, left, right) of the boxes containing the first concept and the response are to be connected. The midpoints of these sides are calculated.

7. A line is drawn between the boxes.

8. For the ISA predicate, "is a" is printed beside the line.

9. For the DOES predicate, the name of the action ("eat", "chase") performed by the agent is placed beside the line.

10. The program processes the next input.

CONCLUSION

The program is a simple illustration of the kind of things that intelligent graphics might bring to computer-assisted learning. It embodies in a simple way some of the rules that might characterize a more fully-developed syntax of graphics. It exemplifies, through equally simple semantic rules, how information in a database might be meanigfully mapped onto syntactically legal graphic structures. However, it does not yet have an instructional component that would correspond to the pragmatic aspect of a complete grammar of graphics. To achieve this, attention must be given to the particular outcomes of instruction and the conditions under which it is to occur, with particular emphasis on student characteristics. In other words, the program needs to have a model of the learner to work from when the form in which onformation is to be presented.

We mentioned earlier that grammars are first and foremost descriptive of usage. This has implications for any further research

that is conducted into the further development of a grammar of graphics for implementation in ICAI. It is not sufficient to conduct experimental studies of how different graphc formats help or hinder understanding. We have an immensely rich graphic tradition in our culture which is characterized more by informal conventions rather than by formal rules (Salomon, 1979, p.20). These conventions, primarily of non-notational systems, need detailed study with a view to making them more formal for application in notational systems, such as charts and diagrams. This represents an early phase in research whose aim is more the identification of possible hypotheses rather than hypothesis testing and requires an exhaustive study of graphics that are part of that tradition (see MacDonald-Ross, 1986, whose study of the historical development of chess manuals exemplifies this type of research). Experimental research can come later.

Let us not lose sight of the reason for the development of a grammar of graphics. While the effective graphics designer does not need to be consciously aware of a "grammar", any more than the successful novelist does, the computer needs to be programmed very specifically to deal with what is intuitively obvious to human graphic artists. Its potential for flexibility and adaptability to student needs (the "conditions" of instruction) relies on its ability to select strategies, including formats in which to present information, during interaction with a student. The computer needs to be able to "doodle" an explanatory graphic on the spur of the moment just as a teacher does. It is therefore no longer sufficient to build predrawn graphic displays into CAI programs. The computer can only achieve the flexibility it needs if, instead, it works with a database and a set of syntactic, semantic and pragmatic rules for converting the knowledge it has access to into an infinite variety of forms involving text and graphics. That is the ultimate goal towards which the program described in this paper makes but a small step.

References

Anderson, J.R., Boyle, C.F., and Yost, G. (1985). The Geometry Tutor. Pittsburgh: Carnegie Mellon University, Advanced Computer Tutoring Project.

Anderson, R.C. (1977). The notion of schemata and the educational enterprise. In R.C. Anderson, R.J. Spiro, & W.E. Montague (Eds). Schooling and the acquisition of knowledge. Hillsdale NJ: Lawrence Erlbaum. 415-431.

Berliner, D.C. (1986). In pursuit of the expert pedagogue. Educational Researcher , 15, No. 7, 5-13.

Bloom, B.S. (1984). The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. Educational Researcher, 13, No. 6, 4-16.

Chomsky, N. (1965). Aspects of a theory of syntax. Cambridge MA: MIT Press.

Chomsky, N. (1962). Syntactic structures. S'Gravenhage: Mouton and Co.

Christopher, A. (1985). Artificial intelligence and computer graphics. Computer Graphics World, August, 11-20.

Cronbach, L.J., and Snow, R.E. (1977). Aptitudes and Instructional Methods. New York: Irvington.

Dwyer, F.M. (1978). Strategies for Improving Visual Learning. State College PA.: Learning Services.

Fleming, M.L., and Levie, W.H. (1978). Instructional Message Design: Principles from the Behavioral Sciences. Englewood Cliffs, NJ.: Educational Technology Publications.

Fowler, R. (1971). An instroduction to transformational syntax. New York: Barnes and Noble Inc.

Gagne, R.M., and Briggs, L.J. (1979). Principles of Instructional Design. New York: Holt, Rinehart and Winston, second edition.

Gagne, R.M., and Dick, W. (1983). Instructional psychology. Annual Review of Psychology, 34, 261-295.

Goodman, N. (1968). The Languages of Art. Indianapolis: Hackett.

Harmon, P., and King., D. (1985). Expert Systems. New York: John Wiley.

Hollan, J.D., Hutchins, E.L., & Weitzman, L. (1984). STEAMER: An interactive inspectable simulation-based training system. The AI Magazine, Summer, 15-27.

Holliday, W.G., Brunner, L.L., & Donais, E.L. (1977). Differential cognitive and affective responses to flow diagrams in science. Journal of Research in Science Teaching, 14, 129-138.

Knowlton, J.Q. (1966). On the definition of 'picture'. AV Communication Review, 14, 157-183.

Koran, M.L., & Koran, J. (1980). Interaction of learner characteristics with pictorial adjuncts in learning from science text. Journal of Research in Science Teaching, 17, 477-483.

Kosslyn, S., Simcox, W.A., Pinker, S., & Parkin, L.P. (1983). Understanding charts and graphs: A project in applied cognitive psychology. ERIC Document Reproduction Service, ED 238 687.

Larkin, J. (1985). Understanding, problem representations, and skill in Physics. In J.W. Segal & S.F. Chipman (Eds.). Thinking and Learning Skills, Vol. 2. Hillsdale: Lawrence Erlbaum.

MacDonald-Ross, M. (1986). Diagrams in scientific texts. Presented at the seminar on Illustration and Text, University of Tubingen, West Germany, July.

Neisser, U (1976). Cognition and Reality. San Francisco: Freeman.

Norman, D.A., Rumelhart, D.E. & LNR Research Group. (1975). Explorations in Cognition. San Francisco: Freeman.

Nunan, T, (1983). Countering Educational Design. New York: Nichols Publishing Company.

Rabinowitz, M., & Mandler, J.M. (1983). Organization and information retrieval. Journal of Experimental Psychology: Learning, Memory, and Cognition, 9, 430-439.

Reigeluth, C.M. (1983). Instructional design: What is it and why is it? In C.M. Reigeluth (Ed.), Instructional Design Theories and Models. Hillsdale, NJ.: Lawrence Erlbaum, 3-36.

Richey, R. (1986). The theoretical and conceptual bases of instructional design. London: Kogan Page.

Salomon, G. (1979). Interaction of Media, Cognition and Learning. San Francisco: Jossey Bass.

Salomon, G. (1981). Communication and Education: Social and

Psychological Interaction. Beverly Hills, CA.: Sage.

Salomon, G. (1982). The differential investment of mental effort in learning from different sources. Annual conference of the American Educational Research Association, New York, March.

Salomon, G. (1983). How students decide when to invest more mental effort in learning. Annual conference of the American Educational Research Association, Montreal, April.

Salomon, G. (1983). The differential investment of mental effort in different sources of learning material. Educational Psychologist, 18, 42-50.

Schank, R.C. (1984). The Cognitive Computer: On Language, Learning and Artificial Intelligence. Reading, MA.: Addison-Wesley.

Schank, R.C., and Abelson, R.P. (1977). Scripts, Plans, Goals and Understanding. Hillsdale, NJ.: Lawrence Erlbaum.

Schlichtmann, H. (1985). Characteristic traits of the semiotic system "Map symbolism". The Cartographic Journal, 22, 23-30.

Schoenfeld, A.H. (1983). Beyond the purely cognitive: Belief systems, social cognitions and metacognitions as driving forces in intellectual performance. Cognitive Science, 7, 329-363.

Schon, D. A. (1983). The Reflective Practitioner: How Professionals Think in Action. New York: Basic Books, Inc.

Schunk, D.H. (1984). Self-efficacy perspective on achievement behavior. Educational Psychologist, 19, 48-58.

Sleeman, D., and Brown, J.S. (1982). Intelligent Tutoring Systems. New York: Academic Press.

Szlichcinski, K.P. (1980). The syntax of pictorial instruction. In P.A. Kolers, M.E. Wrolstad, and H. Bouma, (Eds.), Processing Visible Language Volume II. New York: Plenum Press.

Thorndyke, P.W., & Hayes-Roth, B. (1979). The use of schemata in the acquisition and transfer of knowledge. Cognitive Psychology, 11, 82-106.

Tobias, S. (1976). Achievement treatment interactions. Review of Educational Research, 46, 61-74.

Watzlawick, P., Beavin, J.H., & Jackson, D.D. (1967). Pragmatics of human communication: A study of interactional patterns, pathologies

and paradoxes. New York: Norton.

Wertheimer, M. (1938). Laws of organization in perceptual forms in a source book for Gestalt psychology. London: Routledge and Kegan Paul.

Winn, W.D. (1982). The role of diagrammatic representation in learning sequences, identification and classification as a function of verbal and spatial ability. Journal of Research in Science Teaching, 19, 79-89.

Winn, W.D. (1983). Perceptual strategies used with flow diagrams having normal and unanticipated formats. Perceptual and Motor Skills, 57, 751-762.

Winn, W.D. (in press). Instructional design and intelligent systems: Shifts in the designer's decision-making role. Instructional Science.

Winn, W.D. (1986a).Instructional design and self-designing systems. Educational Technology, 26.

Winn, W.D. (1986b). The design and use of instructional graphics. Presented at the seminar Illustration and Text, University of Tubingen, West Germany, July.

ICAI Model

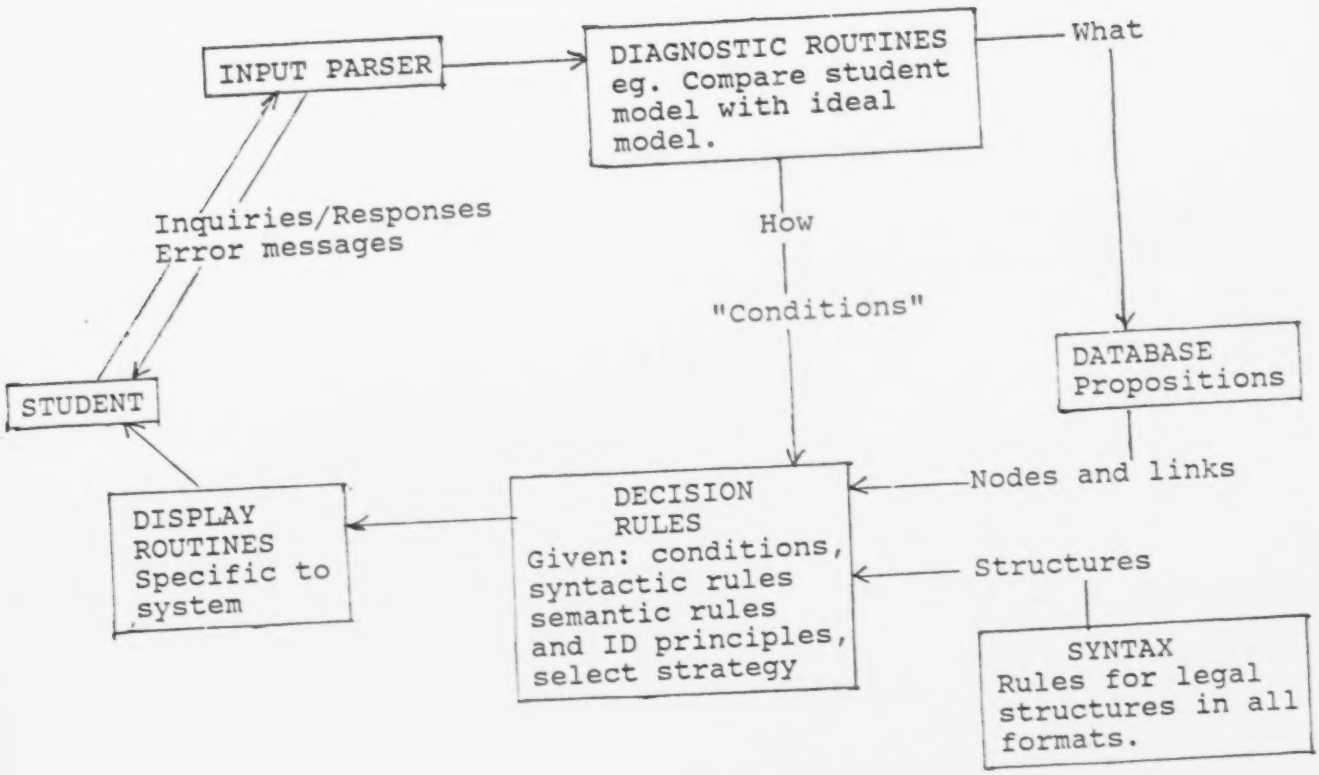

Figure 1

Linguistic Model

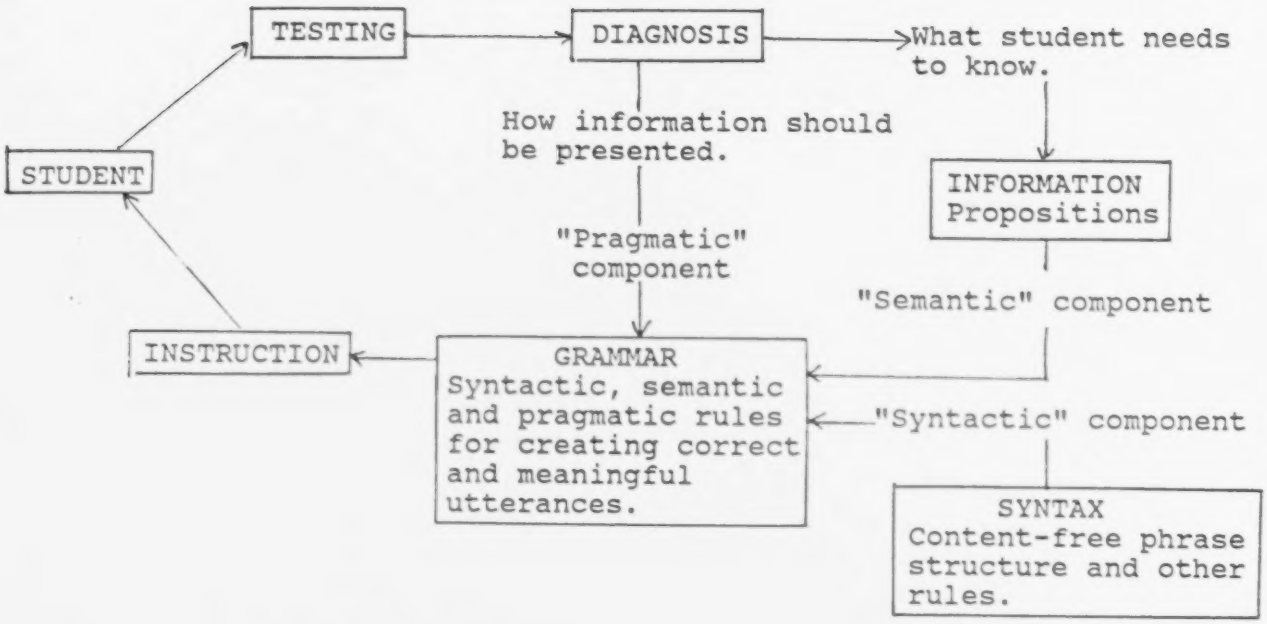

Figure 2

Program Model

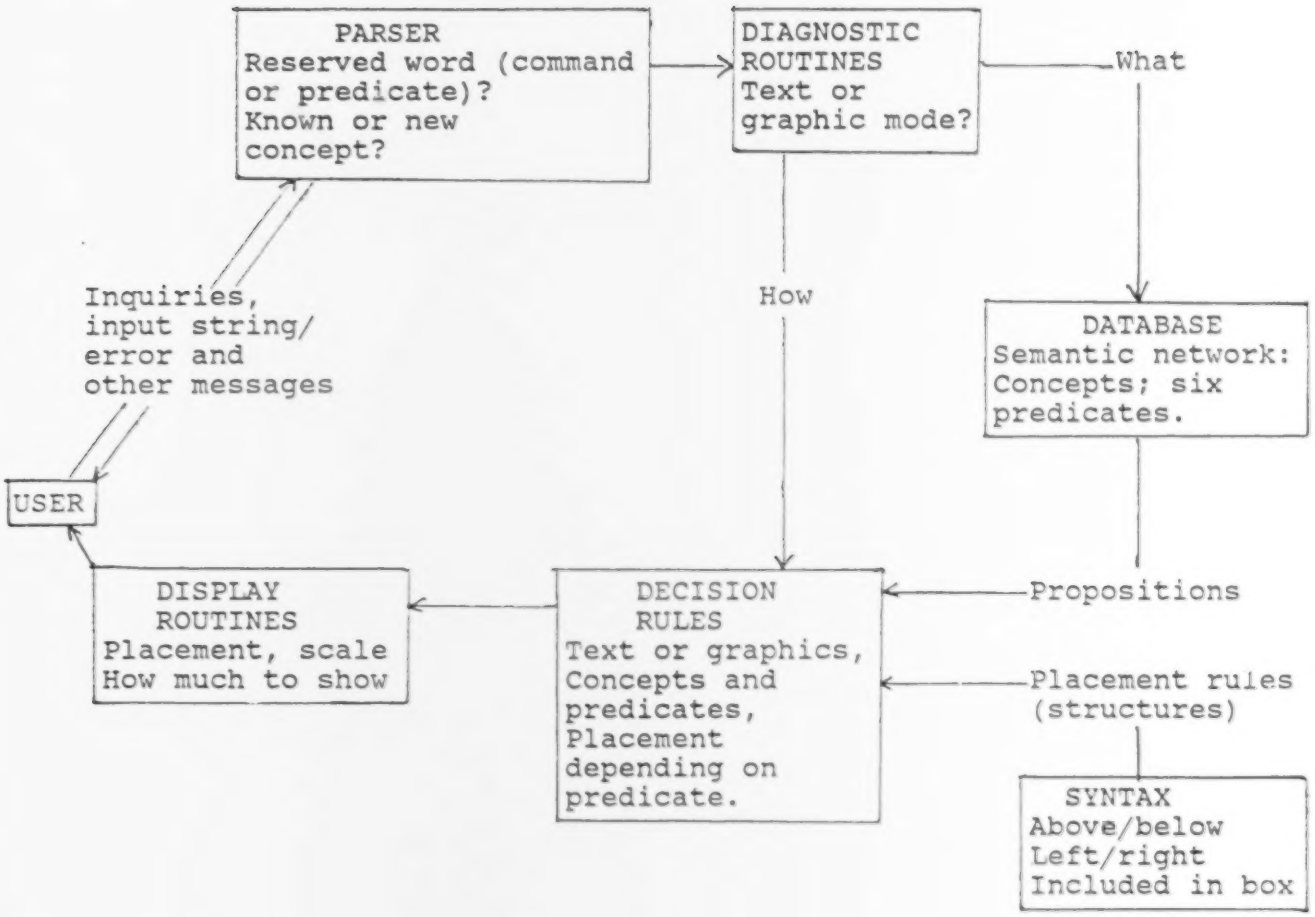

Figure 3